Building Cloud Native, Web Scale, Deployable VNFs with Service Mesh Architecture

Wenjing Chu @ Huawei
Stephen Wong @ Huawei
Rossella Sblendido @ SUSE
Xuan Jia @ China Mobile
Dave Neary @ Red Hat
Isaku Yamahata @ Intel
Motivation

1. Future Telecom Services, e.g. 5G’s top use cases
   - 50 billion IoT devices by 2020
   - Exceptional user experience AR/VR
   - Ultra low latency services (extending cloud to the edge), autonomous vehicles

2. Zero Touch
   - ETSI ZSM
   - “DT: Brutal Automation is Only Way to Succeed” (Lightreading)

3. Application Innovation
   - Monetization with close user engagements
   - Data analytics driven automation and optimization

=> Call for Cloud Native, Web Scale, and Deployable VNFs
Cloud Native is more than containers or Kubernetes

- CNCF defines *cloud native* as
  - Containerized
  - Dynamically Managed by e.g. Kubernetes
  - *Micro-service oriented*

- *Micro-service oriented*
  - The Service Mesh model (e.g. Istio, Conduit...)

Design for Web Scale

- **Scalability**
  - Decouple infrastructure scaling from application development and scaling.
  - Enable traffic splitting, A/B testing, gradual rollout, Canary releases, mirroring etc. sophisticated automated operations.

- **Reliability**
  - Incorporate failure tolerance into the core framework
  - Adopt micro-services: reduce systemic risk, avoid failure cascading
  - Monitor everything: tracing, logging, event monitoring, ... failure as events
  - Implement common operation patterns such as traffic steering, circuit breaker, fault injection, failure isolation (bulkhead), ... for native fault tolerance. Again, failure as events.
  - Aim for 99.999% availability services

=> Service Mesh helps to achieve these goals systematically
Design for Deployability

- **Micro-services with Service Mesh**
  - Are easier to be independently developed right, deployed/operated right
  - Reduce risk, simplify failure modes, enable automation
  - Provide unified toolsets and control
  - Provide uniform security

- **Continuous deployment / DevOps**
  - Design software for operation, remove the developer/operator gap
  - Operate the service for customers, remove the infrastructure operator/service operator/user gap
  - Service Mesh provides systematic policy control

=> Service Mesh helps operators to automate deployment and operation of large scale services
Project Clover: building VNFs with Service Mesh backed micro-services

https://wiki.opnfv.org/display/CLOV

Istio Control Plane API

Control flow during request processing

Discovery & Config data to Proxy

Policy checks, telemetry

TLS certs to Proxy

Pilot
Mixer
Istio-Auth

Pod

Proxy

envoy

svcA

svcB

svcC

svcD

svcE

Service A

Service B

Example VNFs

Common Networking micro-services

DNS DHCP CA Broker

Test Debug

Common cloud native toolset for tracing, logging, monitoring ...

https://wiki.opnfv.org/display/CLOV

CA
Project Clover: NF centric challenges

- Infrastructure:
  - Baremetal, OPNFV Pharos/LaaS, Public Clouds, Network Edge, Edge devices
- Containers and Kubernetes: container4nfv
- Container networking, networking data path
- Cloud native storage and stateful VNFs
- Continuous delivery/deployment on top of CI
- Integration with overall management and operation

=> A cross-project approach in meeting these challenges!
Project Container4NFV: extensions to Kubernetes and container runtime

The goal is to provide container environment where cloud native VNF can run.

- Multus CNI Plugin
- OVS-DPDK
- Istio
- Stor4NFV (Will do)

- Kata Container
- Virtlet
- X86 Platform
- ARM Platform
- SGX (Security)
Container Networking and building VNF data path

Problem description
Container networking: solutions

Routing approach (Calico)

Overlay (Flannel, Docker)

- Agent running on each node
- Uses kernel's L3 forwarding capabilities
- Handles ACLs (iptables, eBPF)
- One of the protocols used to build the Internet
- Used to advertise routes
Cloud native storage and building Stateful VNFs

- Application state changes with microservices
  - Distributed application ➔ Consistency, Availability, Partition-tolerant conflicts
  - Resiliency moves to data layer from hardware layer
  - Back-up, caching strategies must evolve

- Application architecture can evolve
  - Just Enough Duct Tape approach
  - Understanding scaling issue is key
Strategies for scaling out data stores

1. Caching
2. Read-only replicas
3. Alternative data stores
4. Pub/Sub - event driven
Cloud native CI/CD

- CD: Continuous delivery (and deployment)
  - Fast-paced user driven/data driven CI/CD is a core competency for Cloud Native DevOps.

Artifact repo

Commit ➔ Build ➔ Test ➔ Bake ➔ Deploy ➔ Rollback

- A Canary deploy example using Spinnaker.
- A Service Mesh greatly expands this capability.
Integrating with VNF management and operations

- **ONAP**: Container based network service/function deployment
  - ONAP COE as subproject multicloud project

- **Goal/Scope**
  - Allow ONAP orchestrator to handle k8s or other Container orchestration engine in addition to OpenStack(or VM based cloud management system)
  - Life cycle management for containerized VNF

- **Challenges**
  - Abstraction mis-match among VM (openstack) and container(k8s)
    - TOSCA
    - Container networking model is quite different from VM based system
    - K8s and service mesh have different abstractions
Putting all together: A/B Testing

1. CI/CD deploys L7 proxy version 2
2. Istio policy applies for 80% traffic to control (v1) and 20% to variant (v2)
3. Clover software gathers logging / tracing / monitoring and state info to validate “success” or “fail” during a time of traffic
   If success, Istio policy of moving 100% traffic to v2 is applied
4. Clover software gathers info to validate 100% traffic to v2, and results met “success” criteria
Key takeaways

1. Embrace Cloud Native
2. Adopt Micro-services supported by a Service Mesh to replace Monolithic systems
3. Adopt CI/CD end-to-end for a User Driven rapid iterative lifecycle
4. Meet technical challenges together in the open source community
Get involved

- Clover Project
  - [https://wiki.opnfv.org/display/CLOV](https://wiki.opnfv.org/display/CLOV)
  - Fraser demo and Gambia release planning during OPNFV Unconference time
  - Time: Tuesday, March 27, 10:15 - 11:30 AM (Room: K-Town)

- Demo:
  - Huawei booth

- Talk: “Extending CI/CD to support Cloud Native VNFs and Operations: A proposal to the community for discussion”

- Other projects
  - Container4nfv: [https://wiki.opnfv.org/display/OpenRetriever](https://wiki.opnfv.org/display/OpenRetriever)